

# Real-Time Log Viewer System - High Level Design

## 1. System Overview

A web-based system that displays the last N log lines from a target file. It uses Django with Django Channels for WebSocket support and Watchdog to monitor file changes.

## 2. Components

- Frontend: HTML + JS (WebSocket client)
- Backend: Django + Channels (WebSocket handler)
- Watchdog: Detects log file changes
- aiofiles: Asynchronous file reading

## 3. API Endpoints

GET /log/ --> Renders the frontend HTML

WS /ws/fetch\_logs/ --> WebSocket route to stream logs

## 4. Sequence Diagram

User: GET /log/

Django View: Serves index.html

Frontend (JS): WebSocket Connect

LogConsumer: Sends last 10 lines

LogObserver: Watches log file

LogFileHandler: on\_modified triggers

LogConsumer: Sends updated lines

## 5. Key Classes and Functions

- LogConsumer (AsyncWebsocketConsumer): Handles WebSocket connections, sends log lines
- LogObserver: Uses Watchdog to observe file changes
- LogFileHandler (FileSystemEventHandler): Calls consumer's update method
- Views: Renders index.html for log view
- routing.py: Maps WebSocket URLs